

DDL

(Data Definition Language)

Příkazy DDL

- Příkazy pro definování/změnu/mazání struktury dat, např. tabulek, indexů, sekvencí, pohledů a jiných databázových objektů:
 - **CREATE** – vytváření nových objektů (CREATE TABLE, CREATE INDEX, CREATE SEQUENCE, ...),
 - **ALTER** – změna existujících objektů,
 - **DROP** – mazání existujících objektů.

Názvy

- Názvy databázových objektů (název tabulky, ale i sloupce v tabulce) mohou mít maximálně 30 znaků.
- Standardně nejsou názvy case sensitive.
- V případě víceslovného názvu se v názvu databázového objektu používá podtržítko (například UCASTNIK_LETU)
- V názvu databázového objektu je možné mít mezeru, ale pak ho musíte při jakémkoli použití obalit do uvozovek. Uvozovky také ruší case sensitivitu:

```
create table "Zaměstnanci firmy" (jmeno varchar(255));
```

```
select * from "Zaměstnanci firmy";
```

- Něco takového nepoužívejte, jenom si zbytečně přiděláte práci.

Příkazy DDL

- Příkazy CREATE, ALTER, DROP existují pro tyto databázové objekty:
 - INDEX,
 - ROLE,
 - ROLLBACK SEGMENT,
 - SEQUENCE,
 - SYNONYM,
 - TABLE,
 - USER,
 - VIEW aj.
- Nejčastěji používané příkazy budou popsány v této nebo dalších přednáškách.

TEMPORARY TABLE

- Můžete také vytvořit dočasnou tabulku. Definice této tabulky je dostupná ve všech session, ale data jsou v ní viditelná pouze po dobu platnosti jedné konkrétní session a žádná jiná session je nevidí.
- Tuto tabulku vytvoříte následujícím způsobem:

```
CREATE GLOBAL TEMPORARY TABLE TEMP_TABLE  
(JMENO VARCHAR(255));
```

Velikost a umístění tabulky v Oracle

- Při vytváření libovolného objektu v Oracle se volitelně určuje také *počáteční paměťová velikost* objektu, *velikost, o kterou se bude provádět navyšování paměťové oblasti*, když už původní nebude stačit, a *umístění objektu v databázi*.
- Tyto parametry lze doplnit na konec příkazu CREATE TABLE (pokud se určují pro tabulku):
 - Určení umístění: **TABLESPACE** <název prostoru>
 - Zadává se logický úložný prostor, do kterého bude objekt umístěn. Logické prostory obsahují schémata (pro jednotlivé uživatele), do schémat jsou zařazeny jednotlivé databázové objekty. Logickému prostoru odpovídá fyzický soubor na disku (nejčastěji se používá TABLESPACE USERS, data z něj jsou uložena v souboru USERS.DBF).

Velikost a umístění tabulky v Oracle

- Určení počáteční velikosti paměťové oblasti a velikosti, o kterou bude probíhat postupné zvětšování objektu:

```
STORAGE ([INITIAL <celé číslo> [K|M]]  
[NEXT <celé číslo> [K|M]])
```

- Celé číslo za INITIAL udává počáteční velikost pro nově vytvářený objekt (v KB nebo MB).
- Celé číslo za NEXT udává velikost, o kterou se bude paměťová oblast zvětšovat (v KB nebo MB).
- Příklad: STORAGE (INITIAL 20M NEXT 100K).

Speciální verze CREATE TABLE v Oracle

- V Oracle existují další speciální verze příkazu **CREATE TABLE**, které lze použít:

```
CREATE TABLE <název tabulky>  
[( <specifikace sloupce>[, ...])] AS <příkaz SELECT>
```

- Vytvoří tabulku a rovnou ji naplní daty pomocí **SELECTu**. Pokud nejsou specifikovány sloupce vytvářené tabulky, jsou sloupce automaticky vytvořeny na základě sloupců, které vybírá **SELECT** – jednoduše bez integritních omezení.

Příkaz DROP TABLE

- Odstraní z databáze tabulku s daným jménem

`DROP TABLE <název tabulky>;`

- Tabulka bude odstraněna i s případnými indexy, které jsou pro ni definovány, a se všemi integritními omezeními (včetně triggerů navázaných na operace INSERT, UPDATE, DELETE prováděné na tabulce).
- Bude zrušena struktura tabulky, a tudíž i všechna data v tabulce!
- Odstraněním tabulky se stanou nepoužitelnými všechny pohledy vytvořené nad touto tabulkou a procedury/funkce pracující s tabulkou.

Příkaz ALTER TABLE

- Přidání sloupců a integritních omezení do tabulky:

```
ALTER TABLE <název tabulky> ADD (<specifikace sloupce>[, ... ])
```

<specifikace sloupce> je stejná jako u příkazu **CREATE TABLE**

- Úprava sloupců a/nebo integritních omezení v tabulce:

```
ALTER TABLE <název tabulky> MODIFY (<specifikace sloupce>[, ... ])
```

- Odstranění sloupce z tabulky:

```
ALTER TABLE <název tabulky> DROP COLUMN <název sloupce>
```

- Přejmenování sloupce tabulky:

```
ALTER TABLE <název tabulky> RENAME COLUMN <název sloupce>  
TO <nový název sloupce>
```

- Přidání/zrušení/zapnutí/vypnutí integritního omezení v tabulce:

```
ALTER TABLE <název tabulky> {ADD|DROP|ENABLE|DISABLE} CONSTRAINT  
<název omezení> [<definice nového omezení - jen při ADD>]
```

Příkaz ALTER TABLE – příklady

- **ALTER TABLE** dodavatel **ADD** mesto **VARCHAR2**(50);
- **ALTER TABLE** dodavatel **ADD** (
 mesto **VARCHAR2**(50),
 telefon **VARCHAR2**(13)
);
- **ALTER TABLE** dodavatel **MODIFY** mesto **VARCHAR**(64);
- **ALTER TABLE** dodavatel **DROP COLUMN** mesto;
- **ALTER TABLE** dodavatel **RENAME COLUMN** mesto **TO** sidlo;

Příkaz ALTER TABLE – příklady

- **ALTER TABLE** dodavatel **ADD CONSTRAINT** telefon_chk **CHECK** (telefon **LIKE** '+420%');
- **ALTER TABLE** dodavatel **DISABLE CONSTRAINT** telefon_chk;
- **ALTER TABLE** dodavatel **ENABLE CONSTRAINT** telefon_chk;
- **ALTER TABLE** dodavatel **DROP CONSTRAINT** telefon_chk;
- **ALTER TABLE** dodavatel **ADD CONSTRAINT** adresa_fk **FOREIGN KEY** (adresa_id) **REFERENCES** adresa(adresa_id);
- **ALTER TABLE** dodavatel **MODIFY** adresa_id **NUMBER**(10,0) **CONSTRAINT** adresa_fk **REFERENCES** adresa(adresa_id);

Změna typu sloupce

- Změnu typu sloupce nelze u ne zcela kompatibilních typů provést v jednom kroku (např. VARCHAR2 na CLOB), ale provádí se následujícími čtyřmi kroky:
 - Do tabulky přidáme nový sloupec s libovolným dočasným jménem a požadovaným cílovým typem,
 - pomocí UPDATu uložíme do nového sloupce hodnoty z původního sloupce,
 - z tabulky odstraníme původní sloupec,
 - nový sloupec přejmenujeme, aby se jmenoval stejně jako původní sloupec (pokud mu nechceme dát jiné jméno).

Změna typu sloupce – příklad

- Změníme typ sloupce description z VARCHAR2 na CLOB:

```
ALTER TABLE CRM.customer ADD description_new CLOB;  
UPDATE CRM.customer SET description_new = description;  
ALTER TABLE CRM.customer DROP COLUMN description;  
ALTER TABLE CRM.customer RENAME COLUMN description_new  
to description;
```

- Volitelně přidáme k novému sloupci také komentář, který se převodem ztratí:

```
COMMENT ON COLUMN CRM.customer.description  
IS 'Strucny popis zakaznika';
```

READ ONLY TABULKA

- Můžete na tabulce nastavit, že data v ní budou pouze READ ONLY následujícím způsobem:

```
ALTER TABLE ro_products READ ONLY;
```

SYNONYMUM

- Synonymum (SYNONYM) je alias pro jiný databázový objekt. Jsou dva typy synonym:
 - Public synonyma jsou viditelné pro všechny uživatele.
 - Private synonyma jsou viditelné pouze pro vlastníka.
- Synonymum může ukazovat na tabulku, view, sekvenci, proceduru, funkci nebo balíček.
- Synonyma se používají ke zjednodušení SQL.
- Vytvoření private synonyma:
 - **CREATE SYNONYM** [schema.]nazev_synonyma
 - **FOR** [schema.]objekt
- Vytvoření public synonyma:
 - **CREATE PUBLIC SYNONYM** [schema.]nazev_synonyma
 - **FOR** [schema.]objekt